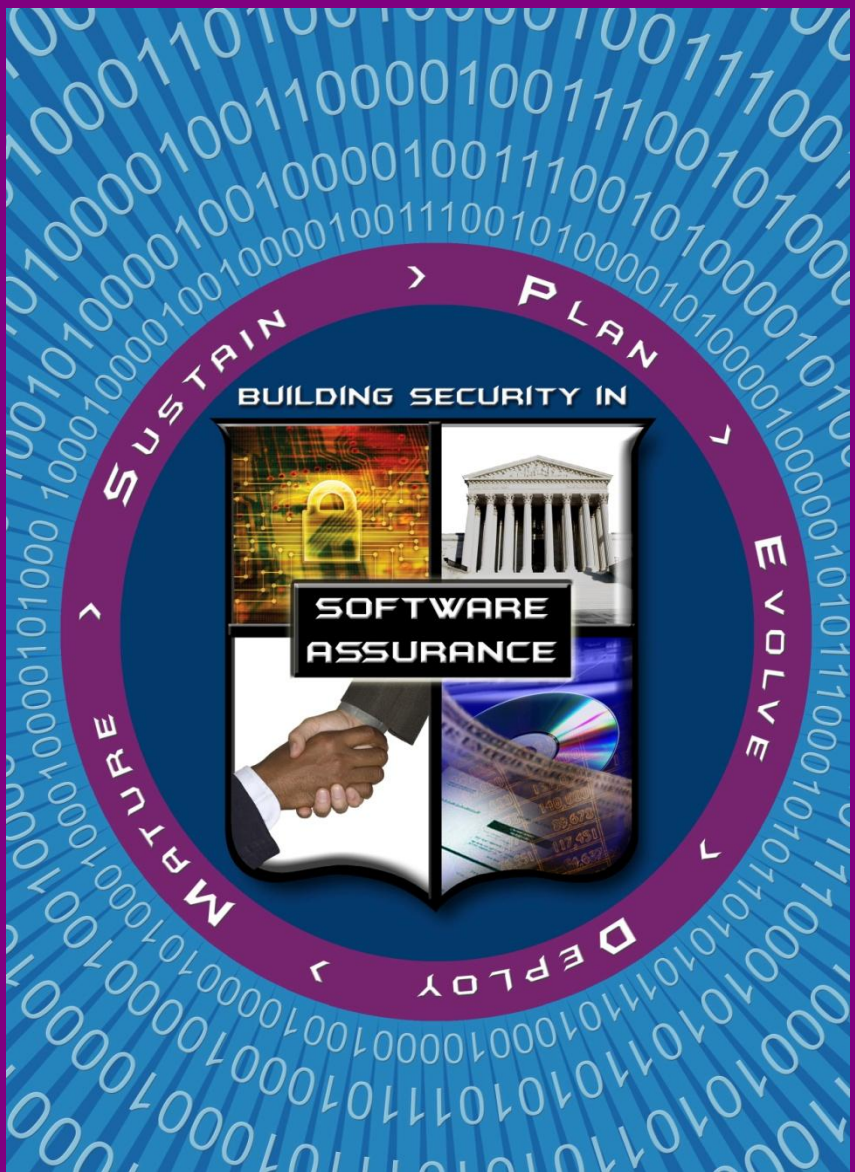
Software Assurance in Education, Training & Certification

Software Assurance Pocket Guide Series:
Life Cycle Support, Volume I
Version 0.5, October 30, 2009 (Draft)



Software Assurance (SwA) Pocket Guide Resources

This is a resource for 'getting started' in educating, training and certifying a workforce with regards to their awareness about the engineering activities and knowledge areas in building software that is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its life cycle, and that the software operates as expected. As part of the Software Assurance (SwA) Pocket Guide series, this resource is offered for informative use only; it is not intended as directive or presented as being comprehensive since it references and summarizes material in the source documents that provide detailed information. When referencing any part of this document, please provide proper attribution and reference the source documents, when applicable.

This volume of the SwA Pocket Guide series focuses on enumerating education, training and certification resources. It identifies the most effective strategies to inject software assurance topics into existing college curriculums and workforce training and certification programs.

At the back of this pocket guide are references, limitation statements, and a listing of topics addressed in the SwA Pocket Guide series. All SwA Pocket Guides and SwA-related documents are freely available for download via the SwA Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa.

Acknowledgements

SwA Pocket Guides are developed and socialized by the SwA community as a collaborative effort to obtain a common look and feel and are not produced by individual entities. SwA Forum and Working Groups function as a stakeholder meta-community that welcomes additional participation in advancing software security and refining. All SwA-related information resources that are produced are offered free for public use. Inputs to documents for the online resources are invited. Please contact Software.Assurance@dhs.gov for comments and inquiries. For the most up to date pocket guides, check the website at https://buildsecurityin.us-cert.gov/swa/.

The SwA Forum and Working Groups are composed of government, industry, and academic members and focuses on incorporating SwA considerations in the acquisition and development processes relative to potential risk exposures that could be introduced by software and the supply chain.

Participants in the SwA Forum's Workforce Education and Training Working Group contributed to developing the material used in this pocket guide as a step in raising awareness on how to incorporate SwA topics in education,

training and certification of a workforce that is knowledgeable to perform engineering activities or aspects of activities relevant for promoting software assurance throughout the Software Development Life Cycle (SDLC).

Information contained in this pocket guide is primarily derived from the documents listed in the *Resource* boxes that follow throughout this pocket guide.

Special thanks go to Assistant Professor Robin A. Gandhi, Ph.D., at the University of Nebraska at Omaha, for providing the synthesis and organization of the material, as well as and the Department of Homeland Security (DHS), National Cyber Security Division's Software Assurance team who provided much of the support to enable the successful completion of this guide and related SwA documents.

Resources

- "Software Assurance: A Curriculum Guide to the Common Body of Knowledge", DHS SwA Forum Workforce Education and Training Working Group, Samuel T. Redwine, Jr. (Editor), Version 1.2, U.S. Department of Homeland Security (DHS), October 2007 at https://buildsecurityin.us-cert.gov/daisy/bsi/940-BSI/version/default/part/AttachmentData/data/CurriculumGuideToTheCBK.pdf.
- "Software Security Assurance: A State-of-the-Art Report", Goertzel, Karen Mercedes, *et al*, Information Assurance Technology Analysis Center (IATAC) of the Defense Technical Information Center (DTIC) at http://iac.dtic.mil/iatac/reports.jsp.
- "Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance," Goertzel, Karen, Theodore Winograd, et al. for Department of Homeland Security and Department of Defense Data and Analysis Center for Software., October 2008 at https://www.thedacs.com/techs/enhanced_life_cycles/.
- NASA Software Assurance Guidebook, at http://sato.gsfc.nasa.gov/guidebook/index.php.
- "IT Security Essential Body of Knowledge (EBK): A Competency and Functional Framework for IT Security Workforce Development," DHS U.S. Computer Emergency Response Team (US-CERT), at http://www.us-cert.gov/ITSecurityEBK/.
- DoD 8570.01-M, "Information Assurance Workforce Improvement Program," Incorporating Change 1, May 15, 2008 Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer, at http://www.dtic.mil/whs/directives/corres/pdf/857001m.pdf.
- "Towards an Organization for Software System Security Principles and Guidelines," version 1.0, Samuel T. Redwine, Jr.,.Institute for Infrastructure and Information Assurance, James Madison University, IIIA Technical Paper 08-01. February 2008 at http://www.jmu.edu/iiia/webdocs/Reports/SwA_Principles_Organization-sm.pdf.
- "Integrating Software Assurance Knowledge Into Conventional Curricula" Crosstalk: The Journal of Defense Software Engineering, Jan 2008, Mead, N.R., Shoemaker, D., & Ingalsbe, J.A., at http://www.stsc.hill.af.mil/crossTalk/2008/01/0801MeadShoemakerIngalsbe.html.

Overview

Current events related to cybersecurity encourage a fundamental shift in the way we think about educating and training a workforce prepared to address security issues in all phases of a software system. Software assurance education and training is aimed to ensure adequate coverage of requisite

Guiding Questions for SwA Curriculum Development:

- *Activities:* What are the engineering activities or aspects of activities that are relevant to achieving secure software?
- *Knowledge:* What knowledge is needed to perform these activities or aspects?

knowledge areas in contributing disciplines such as software engineering (including its many subdisciplines), systems engineering, project management, etc., to identify and acquire competencies associated with secure software. The primary audiences for this pocket guide are educators and trainers who can use this guide to identify resources to supplement their efforts as well as identify strategies to inject software assurance related topics in the existing education and training programs.

The objective of software assurance is to ensure that the processes, procedures, and products used to produce and sustain the software conform to all requirements and standards specified to govern those processes, procedures, and products. Software assurance in its broader sense refers to the assurance of any required property of software. However, in the context of this pocket guide, software assurance is concerned with assuring the security of software.

Building secure software requires a workforce that understands the processes and technologies necessary to provide the basis for belief that software will consistently exhibit all properties required to ensure that the software will operate as expected; despite the presence of faults introduced by a malicious adversary. The Ware Report (1969) identified that:

> "*Probably the most serious risk in system software is **incomplete design**, in the sense that inadvertent loopholes exist in the protective barriers and have not been foreseen by the designers*."

Later the Anderson Report (1972) clearly established the technical problem to be solved as that of:

> "*…determining what constitutes an appropriate defense against malicious attack, and then developing hardware and software with the defensive mechanisms **built in***."

Nearly forty years after, as we find ourselves in the midst of a highly interconnected cyber infrastructure the need for a workforce with better skills to **build security in** cannot be emphasized enough. The objective is to enable a workforce competent in managing, designing, implementing and evaluating systems that can enforce security policies and fulfill security expectations. This workforce should be able to develop a well-reasoned and auditable basis for believing that the software will function as expected, i.e. have justifiable arguments to questions such as:

- How secure is your software?
- What is it secure against?
- How does it achieve its security goals?

The Case for Software Assurance Education

Software assurance has become critical because dramatic increases in business and mission risks are now known to be attributable to exploitable software: system interdependence and software dependence has software as the weakest link; software size and complexity obscures intent and precludes exhaustive test; outsourcing and use of un-vetted software supply chain increases risk exposure; attack sophistication eases exploitation; reuse of legacy software interfaced with other applications in new environments introduces other unintended consequences increasing the number of vulnerable targets; and the number of threats targeting software. These all contribute to the increase of risks to software-enabled capabilities and the threat of asymmetric attack. A broad range of stakeholders now need confidence that the software which enables their core business operations can be trusted to perform (even with attempted exploitation).

In their report to the President, Cyber Security: A Crisis of Prioritization (February 2005), in the chapter entitled "Software Is a Major Vulnerability", the President's Information Technology Advisory Committee (PITAC) summed up the problem of non-secure software concisely and accurately:

> "*Network connectivity provides "door-to-door" transportation for attackers, but vulnerabilities in the software residing in computers substantially compound the cyber security problem. As the PITAC noted in a 1999*

report, the software development methods that have been the norm fail to provide the high quality, reliable, and secure software that the Information Technology infrastructure requires.

Software development is not yet a science or a rigorous discipline, and the development process by and large is not controlled to minimize the vulnerabilities that attackers exploit. Today, as with cancer, vulnerable software can be invaded and modified to cause damage to previously healthy software, and infected software can replicate itself and be carried across networks to cause damage in other systems. Like cancer, these damaging processes may be invisible to the lay person even though experts recognize that their threat is growing. And as in cancer, both preventive actions and research are critical, the former to minimize damage today and the latter to establish a foundation of knowledge and capabilities that will assist the cyber security professionals of tomorrow reduce risk and minimize damage for the long term.

Vulnerabilities in software that are introduced by mistake or poor practices are a serious problem today. In the future, the Nation may face an even more challenging problem as adversaries - both foreign and domestic – become increasingly sophisticated in their ability to insert malicious code into critical software."

It is clear that to produce, acquire, and sustain secure software, a framework that identifies workforce needs for competencies, leverages sound practices, and guide curriculum development for education and training relevant to software assurance is inevitable. Because software quality assurance and software engineering have evolved bodies of knowledge that do not explicitly address security as a quality attribute, a workforce education and training framework must also identify the integration point of secure software development techniques and practices in the existing programs nationwide.

SwA Curriculum and Training Development Guides

- Software Assurance: A Curriculum Guide to the Common Body of Knowledge", DHS SwA Forum Workforce Education and Training Working Group, Version 1.2, Samuel T. Redwine, Jr. (Editor), U. S. Department of Homeland Security, October 2007.
    - The National Strategy to Secure Cyberspace (AR 2-14, Feb 2003) identified the need for a public private effort to promulgate best practices and methodologies that promote integrity, security, and reliability in software code development, including processes and procedures that diminish the possibilities of erroneous code, malicious code, or trap doors that could be introduced during development. To fulfill this need, the initial focus of Software Assurance education has been for persons with knowledge of software quality assurance and software engineering but not security. The efforts of and discussions among participants from academia, government and industry have resulted in a collection of a comprehensive set of principles and guidelines from the disciplines of software engineering, systems engineering, information system, computer science, safety, security, testing, information assurance, and project management. The Software Assurance: A Curriculum Guide to the Common Body of Knowledge to Produce, Acquire, and Sustain Secure Software (referenced in short as SwA CBK) represents this collection of knowledge.

- "Towards an Organization for Software System Security Principles and Guidelines," version 1.0, Samuel T. Redwine, Jr, Institute for Infrastructure and Information Assurance, James Madison University, IIIA Technical Paper 08-01. February 2008.
    - This guide contains an extensive set of software system security principles and guidelines organized in a logical, in-depth fashion. As well as providing coherence, the structure provides grounds for arguing completeness - at least at the higher levels. Historically, principles and guidelines for software system security have originated variously over thirty-plus years, and their authors have tended to provide flat lists occasionally organized topically, by major life cycle stages, or by the author's judgment of importance. The result was hundreds of items whose relationships to each other were unclear and therefore hard to systematically learn, remember, and teach. This

document provides previously lacking coherence and completeness. This is the first highly organized presentation of such a comprehensive set of principles and guidelines. Its structure emphasizes how they relate to each other. The organization aims to start with the most basic, abstract, or inclusive ones and recursively identify the ones that are logically subordinate to each - generally as parts, partial solutions, or causes of them. Thus, it aims to begin to bring needed coherence and intellectual manageability to the area.

- "Software Security Assurance: A State-of-the-Art Report", Goertzel, Karen Mercedes, *et al*, Information Assurance Technology Analysis Center (IATAC) of the DTIC.
  - This state-of-the-art report (SOAR) identifies the current "state-of-the-art" in software security assurance. It provides an overview of the current state of the environment in which defense and national security software must operate; then provides a survey of current and emerging activities and organizations involved in promoting various aspects of software security assurance; and describes the variety of techniques and technologies in use in government, industry, and academia for specifying, acquiring, producing, assessing, and deploying software that can, with a justifiable degree of confidence, be said to be secure. Finally, the SOAR presents some observations about noteworthy trends in software security assurance as a discipline.

- "Enhancing the Development Life Cycle to Produce Secure Software: A Reference Guidebook on Software Assurance," Goertzel, Karen, Theodore Winograd, et al. for Department of Homeland Security and Department of Defense Data and Analysis Center for Software., October 2008.
  - This guide is intended to complement Software Security Assurance: A State-of-the-Art Report, by describing in greater technical depth and detail the security principles and practices that software developers, testers, and integrators can adopt to achieve the twin objectives of producing more secure software-intensive systems, and verifying the security of the software they produce.

- "IT Security Essential Body of Knowledge (EBK): A Competency and Functional Framework for IT Security Workforce Development," DHS US-CERT
  - The IT Security EBK conceptualizes IT security skill requirements in a new way to address evolving IT security challenges. The EBK characterizes the IT security workforce and provides a national baseline representing the essential knowledge and skills that IT security practitioners should have to perform specific roles and responsibilities. As a single foundation linking competencies to security roles, the IT Security EBK will help ensure that we have the most qualified and appropriately trained IT security workforce possible

- DoD 8570.01-M, "Information Assurance Workforce Improvement Program," Incorporating Change 1, May 15, 2008, Assistant Secretary of Defense for Networks and Information Integration/Department of Defense Chief Information Officer.
  - This manual provides guidance and procedures for the training, certification, and management of the DoD workforce conducting Information Assurance (IA) functions in assigned duty positions. It also provides a list of DoD approved certifications that personnel performing IA functions may use to meet baseline requirements. DoD Components may choose any of the approved certifications to meet the applicable certification requirements for each associated IA technical workforce levels.

Strategies for Injecting SwA Knowledge Areas in existing Education and Training Programs

- Mead, N.R., Shoemaker, D., & Ingalsbe, J.A., "Integrating Software Assurance Knowledge Into Conventional Curricula" Crosstalk: The Journal of Defense Software Engineering, Jan 2008 at http://www.stsc.hill.af.mil/crossTalk/2008/01/0801MeadShoemakerIngalsbe.html.

  - The CBK assumes the following:
    "... software assurance is not a separate profession. What is not clear, however, is the precise relationship between the elements of the CBK and the curricula of each potentially relevant field.

    So, the challenge is to correctly integrate secure software assurance practices into each contributing discipline."

Figure 1 – A Possible SwA Curriculum Injection Example

- Different themes under which Graduate or undergraduate courses can be designed
  - Computer Science and Information Assurance Graduate Programs (Examples)
  - Management of Information Science Graduate Programs (Examples)
  - Executive MBA programs for Project Managers (Examples)
- Possible ways to deliver and consume content: Assignments, Projects, Problem Based Learning scenarios
- Links to existing course curriculums and universities with Software Assurance courses.
- Online Learning Opportunities
  - University of Memphis (Act Online) https://www.act-online.net/
    Free cyber security courses that can be leveraged for free
  - SAFE Code tutorials and resources

SwA Tools in Education and Training

A table of tools that can be used in class to provide hands-on experience with SwA Concepts.

Table 1 – Tools for hands-on classroom experience with SWA Concepts

Tool Name	Tool Description	Possible Classroom Uses
ASCE	ASCE supports the key assurance case notations: Goal Structuring Notation and Claims-Arguments-Evidence. Academic license available upon request at http://www.adelard.com/web/hnav/ASCE/index.html.	Assurance case documentation for class assignments and projects, Demonstration of worked examples used on real projects.
Burp Suite	Burp Suite is an integrated platform for attacking web applications. Located at http://www.portswigger.net/suite/	Burp Suite allows to combine manual and automated techniques to enumerate, analyze, scan, attack and exploit web applications
Pharos	Paros is an open source proxy that traps all HTTP and HTTPS data between server and client, including cookies and form fields, which can be intercepted and modified. Located at http://parosproxy.org/index.shtml	Paros can be used as an introduction to web application security assessment.
CERT Secure Coding Standards	Secure coding standards for commonly used programming languages such as C, C++ and Java. Located at https://www.securecoding.cert.org	Online reference; examples of coding do's and don't's
SDMetrics	Analyze the structural properties of UML models using object-oriented measures of design size, coupling, and complexity. Located at http://www.sdmetrics.com/	Examine object-oriented metrics and measures for design and source code artifacts
Splint	Splint is a tool for statically checking C programs for security vulnerabilities and coding mistakes. Located at http://www.splint.org/	Static analysis code checking activities
FindBugs™	A program which uses static analysis to look for bugs in Java code at http://findbugs.sourceforge.net/.	Scan java code repositories for bugs; Introduction to static code checking activities.
Vine	Provides an intermediate language that x86 code can be translated to for Static analysis. Located at http://bitblaze.cs.berkeley.edu/vine.html	Identify data flows analysis binary analysis
Olly Debug	OllyDbg is a 32-bit assembly level debugger for Microsoft Windows. Located at www.ollydbg.de/	Emphasize binary code analysis and particularly useful in cases where source is unavailable. Explain Buffer Overflows
SAMATE Reference Dataset	The purpose of the SAMATE Reference Dataset (SRD) is to provide users, researchers, and software security assurance tool developers with a set of known security flaws. This will allow end users to evaluate tools and tool developers to test their methods. Located at http://samate.nist.gov/index.php/Main_Page.html.	A reference data set can be used in class to reflect upon known flaws in software.

Workforce Certification Options

Certification and Training Opportunities

Table 2 – Workforce Certification Possibilities

Certification Authority	SwA Relevant Certificates	Resources
GIAC - Global Information Assurance Certification	» GIAC Secure Software Programmer - C (**GSSP-C**) » GIAC Secure Software Programmer - .NET (**GSSP-NET**) » GIAC Secure Software Programmer - Java (**GSSP-JAVA**) » GIAC Web Application Penetration Tester (**GWAPT**) » GIAC Certified Penetration Tester (**GPEN**)	http://www.giac.org/certifications/
ISC²	» **CSSLP**CM - Certified Secure Software Lifecycle Professional	http://www.isc2.org/csslp-certification.aspx
EC-Council	» EC-Council Certified Secure Programmer (**ECSP**) » Certified Secure Application Developer (**CSAD**) » Licensed Penetration Tester (**LPT**)	http://www.eccouncil.org/certification.htm
IEEE Computer Society	» Certified Software Development Professional (**CSDP**)	http://www.computer.org/portal/web/certification

Other SwA Education and Training Topics

According to the Sans Institute, one of the Top 20 coolest careers is **Security-Savvy Software Developer**. The security-savvy software developer leads all developers in the creation of secure software, implementing secure programming techniques that are free from logical design and technical implementation flaws. This expert is ultimately responsible for ensuring customer software is free from vulnerabilities that can be exploited by an attacker.

Why It's Cool?

- "You get to make something that actually runs and does something (and won't break under pressure)."
- "These guys are the senior developers by virtue of their programming prowess."

How It Makes a Difference?

- "No security architecture or policy can compensate for poorly written, buggy, insecure software. If one pays the necessary attention to security when a product is initially developed, one doesn't need to go back and add security later on."
- "This is where the rubber meets the road. These are the people making a difference where it really matters...in the software that runs the world."

How to Be Successful?

The role of security-savvy software developer is challenging and rewarding from multiple perspectives. To be successful, you must understand a multitude of attack vectors used to exploit software to avoid the introduction of flaws. This experience is also needed to leverage the same attack tools and techniques an adversary might use to exploit your software, identifying flaws to be addressed before product shipment.

In a development role, your position will be vital to the company's success, including your ability to communicate the techniques used for secure software development to your peers. This can be challenging, since few enjoy having their work criticized and flaws identified, but is a necessary component of an overall secure software strategy. This role is critical to not only the success of the company, but also to all the customers who implement your software. Secure software development has a direct and undeniable impact on the ability of an organization to protect their systems and information assets, and you play a key role in that success.

Details about recommended courses can be found at http://www.sans.org/20coolestcareers/#job18.

Other Resources

» Academic Curricula Samples

- » Carnegie Mellon University CS curriculum at http://www.csd.cs.cmu.edu/education/bscs/index.html#curriculum.
- » University of California at Davis CS curriculum at http://www.cs.ucdavis.edu/courses/exp_course_desc/index.html.
- » George Washington University CS curriculum at http://www.cs.ucdavis.edu/courses/exp_course_desc/index.html
- » Master's program in Secure Software Systems at James Madison University Article about the program at https://buildsecurityin.us-cert.gov/swa/downloads/JMU_SSE.pdf. Program website at http://www.cs.jmu.edu/sss/.
- » Massachusetts Institute of Technology EECS Undergraduate Program at http://www.eecs.mit.edu/ug/index.html.
- » Stanford University CS curriculum at http://cs.stanford.edu/Courses/.
- » University of Virginia CS curriculum at http://www.cs.virginia.edu/classes/index.php.
- » Virginia Tech CS curriculum at http://www.cs.vt.edu/home/courses.html.

» Commercial Training Examples

- » Aspect Security, Inc., Application Security Education and Training at http://www.aspectsecurity.com/training.htm.
- » Ciphent Certification Programs, Certified Secure Software Developer and Certified Master Secure Software Developer at http://www.ciphent.com/training/certification.
- » Foundstone, Inc., Education at http://www.foundstone.com/us/education-overview.asp.
- » KRvW Associates, LLC., Training Services at http://www.krvw.com/training/training.html.
- » LogiGear, Inc., Web and Software Application Security Testing at http://www.logigear.com/training/course_catalog/course.asp?courseId=20.
- » Microsoft Corp., Clinic 2806: Microsoft® Security Guidance Training for Developers (and other courses) at https://www.microsoftelearning.com/eLearning/courseDetail.aspx?courseId=26043.
- » Netcraft, Inc., Web Application Security Course at http://audited.netcraft.com/web-application-course.
- » Next Generation Security Software, Ltd., Security Training at http://www.ngssoftware.com/consulting/training/.
- » Paladion Networks Pvt. Ltd., Training for Software Developers at http://www.paladion.net/paladion.php?id=18.
- » The SANS Institute, Inc. at https://www.sans.org/.
- » Security Innovation, Inc., Application Security Education at http://www.securityinnovation.com/services/education/index.shtml.
- » Symantec Corp., Application Security Principles and Security in Software Development Lifecycle at https://education.symantec.com/Saba/Web/sena.

Conclusion

This pocket guide compiles software assurance education and training resources aimed to ensure adequate coverage of requisite knowledge areas and the corresponding roles in the workforce. In doing so it draws upon contributing disciplines such as software engineering (including its many subdisciplines), systems engineering, project management, etc., to identify and acquire competencies associated with secure software.

The Software Assurance Pocket Guide Series is developed in collaboration with the SwA Forum and Working Groups and provides summary material in a more consumable format. The series provides informative material for SwA initiatives that seek to reduce software vulnerabilities, minimize exploitation, and address ways to improve the routine development, acquisition and deployment of trustworthy software products. Together, these activities will enable more secure and reliable software that supports mission requirements across enterprises and the critical infrastructure.

For additional information or contribution to future material and/or enhancements of this pocket guide, please consider joining any of the SwA Working Groups and/or send comments to Software.Assurance@dhs.gov. SwA Forums are open to all participants and free of charge. Please visit https://buildsecurityin.us-cert.gov for further information.

No Warranty

This material is furnished on an "as-is" basis for information only. The authors, contributors, and participants of the SwA Forum and Working Groups, their employers, the U.S. Government, other participating organizations, all other entities associated with this information resource, and entities and products mentioned within this pocket guide make no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose, completeness or merchantability, exclusivity, or results obtained from use of the material. No warranty of any kind is made with respect to freedom from patent, trademark, or copyright infringement. Reference or use of any trademarks is not intended in any way to infringe on the rights of the trademark holder. No warranty is made that use of the information in this pocket guide will result in software that is secure. Examples are for illustrative purposes and are not intended to be used as is or without undergoing analysis.

Reprints

Any Software Assurance Pocket Guide may be reproduced and/or redistributed in its original configuration, within normal distribution channels (including but not limited to on-demand Internet downloads or in various archived/compressed formats).

Anyone making further distribution of these pocket guides via reprints may indicate on the pocket guide that their organization made the reprints of the document, but the pocket guide should not be otherwise altered. These resources have been developed for information purposes and should be available to all with interests in software security.

For more information, including recommendations for modification of SwA pocket guides, please contact Software.Assurance@dhs.gov or visit the Software Assurance Community Resources and Information Clearinghouse: https://buildsecurityin.us-cert.gov/swa to download this document either format (4"x8" or 8.5"x11").

Software Assurance (SwA) Pocket Guide Series

SwA is primarily focused on software security and mitigating risks attributable to software; better enabling resilience in operations. SwA Pocket Guides are provided; with some yet to be published. All are offered as informative resources; not comprehensive in coverage. All are intended as resources for 'getting started' with various aspects of software assurance. The planned coverage of topics in the SwA Pocket Guide Series is listed:

SwA in Acquisition & Outsourcing

I. Software Assurance in Acquisition and Contract Language
II. Software Supply Chain Risk Management & Due-Diligence

SwA in Development

I. Integrating Security in the Software Development Life Cycle
II. Key Practices for Mitigating the Most Egregious Exploitable Software Weaknesses
III. Risk-based Software Security Testing
IV. Requirements & Analysis for Secure Software
V. Architecture & Design Considerations for Secure Software
VI. Secure Coding & Software Construction
VII. Security Considerations for Technologies, Methodologies & Languages

SwA Life Cycle Support

I. SwA in Education, Training & Certification
II. Secure Software Distribution, Deployment, & Operations
III. Code Transparency & Software Labels
IV. Assurance Case Management
V. Assurance Process Improvement & Benchmarking
VI. Secure Software Environment & Assurance Ecosystem
VII. Penetration Testing throughout the Life Cycle

SwA Measurement & Information Needs

I. Making Software Security Measurable
II. Practical Measurement Framework for SwA & InfoSec
III. SwA Business Case & Return on Investment

SwA Pocket Guides and related documents are freely available for download via the DHS NCSD Software Assurance Community Resources and Information Clearinghouse at https://buildsecurityin.us-cert.gov/swa.